**Login : Jurnal Teknologi Komputer**
**ISSN : 2302-9692 (print) | 2723-8695 (online)**
**Vol. 14, No. 2, December 2020, pp. 78-83**
LOGIN http://login.seaninstitute.org/index.php/Login


# Implementation of Artificial Neural Networks to Predict The Number of Shares in Companies using The Learning Vector Quantization Method

**Yarina Halawa**
Informatics Engineering Study Program, STMIK Pelita Nusantara, Medan, Indonesia.

**Article Info**

***Article history:***

Received Okt 13, 2020
Revised Nov 23, 2020
Accepted Des 29, 2020

***Keywords:***

Artificial Neural Network,
Learning Vektor Quantization (LVQ),
PHP,
MYSQL.

**ABSTRACT**

Stock prediction is a very influential thing for a company, because stocks are relatively stable and even tend to increase and can affect the views of investors from various companies to invest their funds into a company. Therefore it is also important for a company to be able to predict profits and losses suffered by the company accurately. In predicting shares in PT. Sukses Anugrah Sejahtera in general still uses a system that is still manual, thus making data processing to predict stocks still inaccurate. To overcome the above problems by building and implementing an Artificial Neural Network system application using the Learning Vector Quantization method to help predict the number of shares of the company PT. Sukses Anugrah Sejahtera which starts from the prediction table, determines the initial weight, completes the first literacy to the fifth literacy classification / prediction so that it can determine the rise or fall of shares in PT.SAS companies.



***Corresponding Author:***

Yarina Halawa,
Informatics Engineering Study Program,
STMIK Pelita Nusantara, Medan, Indonesia,
Jl. Iskandar Muda No.1, Merdeka, Kec. Medan Baru, Kota Medan, Sumatera Utara 20154.
Email: Halawarina70@gmail.com

## 1. INTRODUCTION

Stock predictions are very influential for a company, because stocks are relatively stable and even tend to increase and can affect the view of an investor from various companies to invest their funds into a company[1], [2]. There fore it is also important for a company to be able to accurately predict the profits and losses experienced by the company[3].

PT. The success of Anugrah Sejahtera is to be able to predict the advantages and disadvantages of the shares they own more accurately, but the stock predictions they do are often constrained because their stock calculations are still done manually. Based on research conducted by researchers, which researchers got from a journal entitled "Learning Vector Quantization Algorithm and Fuzzy K-NN for Prediction of Stocks Based on Competitors"[4], the author can conclude that stock prediction carried out by the Learning Vector Quantization method has much higher accuracy. Compared to the Fuzzy K-NN method, therefore the researcher will conduct this research using the Learning Vector Quantization method. Learning Vector Quantization is one of the Artificial Neural Networks which is a classification method where each output unit presents a class that aims to approach the distribution of the vector class to minimize errors in its classification.

## 2. RESEARCH METHOD

In this study collecting data as material and the completion stage by collecting data, interviews, analysis methods and data analysis using the Learning Vector Quantization method. Learning Vector Quantization Artificial neural networks (ANN) are a branch of AI (Artificial Intelligence) or artificial intelligence[4], [5]. Artificial Neural Network is one of the information processing systems formed to mimic how the human brain works in solving a problem by carrying out the learning process through changes in the weight it receives[6]–[8]. ANN is able to identify past data-based activities or learn from experience. Past data will be studied by an artificial neural network so that it has the ability to make decisions about data that has never been studied[9]–[11].

Artificial Neural Networks model biological neural networks contained in the human brain. Modeling is mainly only approached from the point of rotation. JST consists of a number of nodes (nodes) which are processing elements[12]. Each of these nodes models a biological nerve cell (neuron). Relationships between nodes is achieved through the connection weight (weight). The connection weight determines whether the flowing signal is inhibitory (inhibitory connection). The connection weight which is muting can be expressed, for example by a negative number, while that which is stimulating by a positive number. Apart from being determined by the weight characteristics of the connection, the amount of signal that comes out of a node is also determined by the activation function it uses. That is, the selection of the activation function determines the degree of activity of a node.

## 3. RESULTS AND DISCUSSION

The research was conducted at PT. The success of Anugrah Sejahtera researchers has encountered the problems faced in this institution. The obstacles faced are the large number of shares that must be managed and developed. However, data collection and data management are still done manually.

The analysis that will be built is a prediction system for the number of shares at PT. Sukses Anugrah Sejahtera uses the Learning Vector Quantization Method which is designed using PHP and MYSQL programming.

Before making further predictions, the company data and actual fluctuation data are first obtained, as in the following table :

**Table 1.** Actual Fluctuation Table

| TANGGAL | FLUKTUASI | | | | FLUKTUASI AKTUAL |
|---|---|---|---|---|---|
| | BANK MANDIRI | BANK DANAMON | PT. MUSIMAS | PT. SMART | |
| 01-01-2020 | Turun | Turun | Turun | Turun | Naik |
| 02-01-2020 | Naik | Naik | Naik | Naik | Naik |
| 03-01-2020 | Naik | Naik | Naik | Naik | Naik |
| 04-01-2020 | Turun | Turun | Turun | Turun | Naik |
| 08-01-2020 | Naik | Naik | Naik | Naik | Naik |
| 09-01-2020 | Naik | Turun | Naik | Naik | Turun |
| 10-01-2020 | Turun | Naik | Turun | Naik | Naik |
| 11-01-2020 | Turun | Turun | Turun | Naik | Naik |
| 15-01-2020 | Naik | Naik | Naik | Turun | Turun |
| 16-01-2020 | Naik | Naik | Naik | Naik | Naik |
| 17-01-2020 | Turun | Naik | Turun | Naik | Turun |
| 18-01-2020 | Turun | Turun | Turun | Turun | Naik |
| 21-01-2020 | Turun | Naik | Turun | Turun | Naik |
| 22-01-2020 | Turun | Turun | Turun | Turun | Naik |
| 23-01-2020 | Naik | Turun | Naik | Turun | Naik |
| 24-01-2020 | Turun | Turun | Turun | Naik | Naik |
| 25-01-2020 | Naik | Naik | Naik | Naik | Naik |
| 28-01-2020 | Turun | Turun | Turun | Turun | Turun |

| | | | | | |
|---|---|---|---|---|---|
| 29-01-2020 | Naik | Naik | Naik | Naik | Naik |
| 30-01-2020 | Turun | Turun | Turun | Turun | Naik |

**Table 2.** Prediction Table

| No | X1 | X2 | X3 | X4 | Target |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 | 1 |

The initial weights are: {1,1,1,0} and {1,0,1,1} with a learning rate of 0.05 with a learning function of 0.1.

Training:

**a. 1st iteration**

1st data {0 , 1, 1, 0} with a target of 0, weight = {{1, 1, 1, 0}, {1, 0, 1, 1}}

calculate the weight for each output:

$D_{(Wj,Xn)} = \sqrt{\sum(Wij, Xni)^2}$

Class 0=$\sqrt{(0-1)^2+(1-1)^2+(1-1)^2+(0-0)^2=1}$

Class 1 = $\sqrt{(0-1)^2+(1-0)^2+(1-1)^2+(0-1)^2=1.7320508075689}$

Determine the winning class:

output = min (class 0, class 1) = class 0

update weights:

since target 0 is equal to output 0, then update the weight:

$W_{ij,new} = W_{ij,\ current} + \eta(X_{ni}-W_{ij,current})$

W11 (new) = W11 (old) α (x11- w11 ( old) = 1 + (0.05 * (0 - 1)) = 0.95

W12 (new) = W12 (old) α (x12- w12 ( old) = 1 + (0.05 * (1 - 1)) = 1

W13 (new) = W13 (old) α (x13- w13 ( old) = 1 + (0.05 * (1 - 1)) = 1

W11 (new) = W11 (old) α (x11- w11 ( old) = 0 + (0.05 * (0 - 0)) = 0

then we get new weights = {{0.95, 1, 1, 0}, {1, 0, 1, 1}} to be used in further data calculations

**b. Second data**

{0 , 0, 1, 1} with a target of 1, weight = {{0.95, 1, 1, 0}, {1, 0, 1, 1}} calculate the weight for each output:

Class 0 =$\sqrt{(0-0{,}95)^2+(0-1)^2+(1-1)^2+(0-0)^2=1.7036725037401}$

Class 1 = $\sqrt{(0-1)^2+(0-0)^2+(1-1)^2+(1-1)^2=1}$

determine the winning class:

output = min (class 0, class 1) = class 1

update weights:

because target 1 is equal to output 1, then update the weight:

W21 (new) = W21 (old) α (x21- w21 ( old) = 1 + (0.05 * (0 - 1)) = 0.95

W22 (new) = W22 (old) α (x22- w22 ( old) = 0 + (0.05 * (0 - 0)) = 0

W23 (new) = W23 (old) α (x23- w23 ( old) = 1 + (0.05 * (1 - 1)) = 1

W24 (new) = W24 (old) α (x24- w24 ( old) = 1 + (0.05 * (1 - 1)) = 1

then we get new weights = {{0.95, 1, 1, 0}, {0.95, 0, 1, 1}} to be used in further data calculations

**c. Data to 3**

{1 , 1, 1, 1} with a target of 0, weight = {{0.95, 1, 1, 0}, {0.95, 0, 1.1}}

calculate the weight for each output:

Class 0 =$\sqrt{(1-0{,}95)^2+(1-1)^2+(1-1)^2+(1-0)^2=1.001249219725}$

Class 1 = $\sqrt{(1-0{,}95)^2+(1-0)^2+(1-1)^2+(1-1)^2=1.001249219725}$

determine the winning class:

output = min (class 0, class 1) = class 0,

because the output between the two classes is the same, one can be selected as the output

update weights:

since target 0 is equal to output 0, then update the weight:

W11 (new) = W11 (old) α (x11- w31 ( old) = 0.95 + (0.05 * (1 - 0.95)) = 0.9525
W12 (new) = W12 (old) α (x12- w32 ( old) = 1 + (0.05 * (1 - 1)) = 1
W13 (new) = W13 (old) α (x13- w33 ( old) = 1 + (0.05 * (1 - 1)) = 1
W14 (new) = W14 (old) α (x14- w34 ( old) = 0 + (0.05 * (1 - 0)) = 0.05
then obtained a new weight = {{0.9525, 1, 1, 0.05}, {0.95, 0, 1, 1}} to be used in further data calculations

**d. Data 4**

{1 , 0,0,1 } with target1, weight = {{0.9525,1,1,0.05}, {0.95,0, 1, 1}}
calculate the weight for each output:
Class 0 =$\sqrt{(1-0{,}9525)^2+(0-1)^2+(0-1)^2+(1-0{,}05)^2} = 1.704334547558$
Class 1 = $\sqrt{(1-0{,}95)^2+(0-0)^2+(0-1)^2+(1-1)^2} = 1.001249219725$
determine the winning class:
output = min (class 0, class 1) = class 1
update weights:
because target 1 is equal to output 1, then update the weight:
W21 (new) = W21 (old) α (x21- w41 ( old) = 0.95 + (0.05 * (1 - 0.95)) = 0.9525
W22 (new) = W22 (old) α (x22- w42 ( old) = 0 + (0.05 * (0 - 0)) = 0
W23 (new) = W23 (old) α (x23- w43 ( old) = 1 + (0.05 * (0 - 1)) = 0.95
W24 (new) = W24 (old) α (x24- w44 ( old) = 1 + (0.05 * (1 - 1)) = 1
then obtained a new weight = {{0.9525, 1, 1, 0.05}, {0.9525, 0, 0.95, 1}} to be used in further data calculations After all the data is calculated, the learning rate updates :
new learning rate = learning function * learning rate ;
= 0.1 * 0.05 = 0.005

**e. Classification / Prediction**

After obtaining the weight of the training results, namely { {0.9525 , 1, 1, 0.05}, {0.9525, 0, 0.95, 1}}, then we try to test the classification of two data, namely {1, 1, 1, 0} and { 1, 0, 1, 1}.

1. 1st data {1 , 1, 1, 0}
   Kelas 0=$\sqrt{(1-0{,}9525)^2+(1-1)^2+(1-1)^2+(0-0{,}05)^2} = 0.068965571120$
   Kelas1=$\sqrt{(1-0{,}9525)^2+(1-0)^2+(1-0.95)^2+(0-1)^2} = 1.415894152117$
   output = class 0
2. Data 2 {1 , 0, 1, 1}
   Class 0 =$\sqrt{(1-0{,}9525)^2+(0-1)^2+(1+1)^2+(1-0{,}05)^2} = 1.380129070051$
   Class 1 =$\sqrt{(1-0{,}9525)^2+(0-0)^2+(1-0.95)^2+(1-1)^2} = 0.068965571120$
   output = class 1

**f. Analysis of Learning Vector Quantization**

The final objective of LVQ analysis is to find the appropriate weight value to classify vectors into destination classes that have been initialized at the time of the formation of the LVQ network. While the test algorithm is to calculate the output value (vector class) closest to the input vector, or it can be equated with the classification process (grouping).
The information we use is as follows:

X : training vector (input) (x1,… ..., xi, ……., xn)
Q : the appropriate category or class for training vectors
Wj : vector weight for the jth unit of output (w1j, ……., Wij, ....., wnj)
Cj : the category or class displayed by the jth unit of output
|| x - wj | | : Euclidean distance between the input vector and the weight vector for the jth output layer

The following is the LVQ learning algorithm:

step 0 : initialize the reference vector; learning rating initialization α (0)
Step 1 : When the stop condition is false, perform steps 2 to 6
step 2 : for each vector training input x perform steps 3 - 4
step 3 : find j to || x - wj || minimum

step 4 : update wj as follows:

if T = Cj, then

Wj ( new) = Wj (old) + α [x - wj (old)];

if T ≠ Cj, then

Wj ( new) = Wj (old) - α [x - wj (old)];

Step 5 : Less rating training

Step 6 : test stop condition: that is, the condition that may specify a fixed number of iterations or a learning rating reaching a sufficiently small value.

After the Learning Vector Quantization (LVQ) learning process , the next step is the process of taking the Learning Vector Quantization (LVQ) output . Retrieval of results is carried out only in the Testing process (testing). Basically, this stage only enters the final weight input then looks for the closest distance by calculating the Euclidean (the closest distance).

## 4. CONCLUSION

After calculating each weight value and looking for the vector x value so that it can make it easier to make decisions as an alternative in predicting stocks with the Learning Vector Quantization (LVQ) learning process , the next step is the process of taking the Learning Vector Quantization (LVQ) output . Retrieval of results is carried out only in the Testing process (testing). Basically, this stage only enters the final weight input then looks for the closest distance by calculating the Euclidean (the closest distance).

## REFERENCES

[1] S. E. Ginting and A. Rikki, "Prediction of Sparepart Sales Level using Exponential Smoothing Method," *Login J. Teknol. Komput.*, vol. 12, no. 2, pp. 44–52, 2018, doi: https://doi.org/10.24224/login.v12i2.27.

[2] S. Sulistyono and W. Sulistiyowati, "Peramalan Produksi dengan Metode Regresi Linier Berganda," *PROZIMA (Productivity, Optim. Manuf. Syst. Eng.*, 2018, doi: 10.21070/prozima.v1i2.1350.

[3] A. Rikki, "Pengujian Sistem Pendukung Keputusan Metode Simple Additive Weighting dan Weighted Product dengan Matlab," *MEANS (Media Inf. Anal. dan Sist.*, vol. 2, no. 1, pp. 47–51, 2017.

[4] R. R. Fitriani, Ernastuti, and E. R. Swedia, "ALGORITMA LEARNING VECTOR QUANTIZATION DAN FUZZY K-NN UNTUK PREDIKSI SAHAM BERDASARKAN PESAING," *J. Ilm. Teknol. dan Rekayasa*, 2019, doi: 10.35760/tr.2019.v24i1.1929.

[5] I. G. S. E. Putra, D. I. K. G. Putra, and P. A. Bayupati, "Klasifikasi Sidik Jari Dengan Metode Fuzzy Learning Vector Quantization Dan Fuzzy Backpropagation," *Semin. Nas. Sist. Inf. Indones. 22 Sept. 2014*, 2014.

[6] A. N. Khomarudin and L. Efriyanti, "PENGEMBANGAN MEDIA PEMBELAJARAN MOBILE LEARNING BERBASIS ANDROID PADA MATA KULIAH KECERDASAN BUATAN," *J. Educ. J. Educ. Stud.*, 2018, doi: 10.30983/educative.v3i1.543.

[7] H. A. Ramadhan and D. A. Putri, "Big Data, Kecerdasan Buatan, Blockchain, dan Teknologi Finansial di Indonesia," *Direktorat Jenderal Apl. Inform. Kementeri. Komun. dan Inform.*, 2018.

[8] M. Abdi, D. Herumurti, and I. Kuswardayan, "Analisis Perbandingan Kecerdasan Buatan pada Computer Player dalam Mengambil Keputusan pada Game Battle RPG," *JUTI J. Ilm. Teknol. Inf.*, 2017, doi: 10.12962/j24068535.v15i2.a671.

[9] A. S. Ritonga and S. Atmojo, "Pengembangan Model Jaringan Syaraf Tiruan untuk Memprediksi Jumlah Mahasiswa Baru di PTS Surabaya (Studi Kasus Universitas Wijaya Putra)," *J. Ilm. Teknol. Inf. Asia*, 2018, doi: 10.32815/jitika.v12i1.213.

[10] "Pengenalan Jenis Golongan Darah Menggunakan Jaringan Syaraf Tiruan Perceptron," *J. Masy. Inform.*, 2017, doi: 10.14710/jmasif.v7i1.10794.

[11] A. Sudarsono, "Jaringan Syaraf Tiruan Untuk Memprediksi Laju Pertumbuhan Penduduk

Menggunakan Metode Backpropagation (Studi Kasus di Kota Bengkulu)," *Media Infotama*, 2016.

[12] Alex Rikki, "APLIKASI JARINGAN SYARAF TIRUAN UNTUK PENENTUAN KONSENTRASI PROGRAM STUDI BAGI CALON MAHASISWA BARU STMIK BUDIDARMA MEDAN," vol. II, pp. 1–4, 2012.